

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application

Inventor(s): Kyle Marvin

Appln. No.: 10/776,435

Confirm. No.: 2900

Filed: February 11, 2004

Title: SYSTEMS AND METHODS FOR
A COMMON RUNTIME CONTAINER
FRAMEWORK

PATENT APPLICATION

Art Unit: 2193

Examiner: Vu, Tuan A.

Customer No. 23910

PROPOSED CLAIMS AMENDMENT FOR ENABLING EXAMINER'S AMENDMENT

Dear Sir:

INTRODUCTORY COMMENTS

This proposed claims amendment is for enabling examiner's amendment and also in response to the phone interview between the Examiner Vu, Tuan A. and the Patent Agent Kuiran (Ted) Liu (#60,039) on July 30, 2008 and July 31, 2008.

Amendments to the Claims begin on page 2.

Remarks/Arguments begin on page 8.

In the Claims:

Please amend claims 1, 6, 23, 28, 43, and 48, cancel claims 2-4, 7-22, 24-26, 29-42, 44-46, and 49-64, and add new claims 65-76. All pending claims are reproduced below, including those that remain unchanged.

1. (Currently Amended) A computer-implemented system to provide a common runtime container framework, comprising:

a microprocessor;

a plurality of runtime containers ~~capable of processing~~ operable to process service requests and providing application services, wherein the plurality of runtime containers are organized in a first hierarchical architecture;

a plurality of metadata objects ~~capable of providing~~ operable to provide metadata on context, state, and/or ~~other runtime~~ information about pertinent to the data and objects being ~~processed~~ handled by the plurality of runtime containers, wherein the plurality of metadata objects are organized in a second hierarchical architecture; and

~~wherein each runtime container of the plurality of runtime containers in the first hierarchical architecture can invoke service components within the runtime container and provide state information and context information to the service components based on a meta-object of the plurality of metadata objects in the second hierarchical architecture at a corresponding level~~

wherein the first hierarchical architecture and the second hierarchical architecture are implemented according to Object-Oriented class inheritance, wherein each runtime container object at one layer of the first hierarchical architecture functionally interacts with a metadata object at a corresponding layer in the second hierarchical architecture, the dual interaction between the objects in the first hierarchical architecture and the second hierarchical architecture comprising:

invoking application services within a runtime container in the first hierarchical architecture; and

utilizing services performed by a correspondent metadata object in the second hierarchical architecture,

wherein the service provided by a metadata object in the second hierarchical architecture comprises providing state information and context information of the application runtime objects needed by the runtime container object directed to handle at least one of the service requests,

wherein the metadata object comprises object methods to get and set metadata required by the runtime container object,

wherein the runtime container object comprises object methods to invoke the needed service provided by the methods of the metadata object.

2-4. (Canceled).

5. (Original) The system in claim 1, further comprising:

a well-defined API capable of creating new types of runtime containers, or customizing existing containers with incremental features.

6. (Currently Amended) The system in claim 1, further comprising:

a well-defined API capable of creating new levels in the first hierarchical architecture ~~for the runtime container and the second hierarchical architecture the metadata object.~~

7-22. (Canceled).

23. (Currently Amended) A method to provide a common runtime container framework, comprising:

processing service requests and providing application services via a runtime container of a plurality of runtime containers, wherein the plurality of runtime containers are organized in a first hierarchical architecture;

providing metadata on context, state, and/or other runtime information ~~about~~ pertinent to the data and objects being ~~processed~~ handled by the runtime container via a metadata object of a plurality of metadata objects, wherein the plurality of metadata objects are organized in a second hierarchical architecture; and

~~allowing each runtime container of the plurality of runtime containers in the first~~

~~hierarchical architecture to invoke service components within the runtime container and provide state information and context information to the service components based on a meta-object of the plurality of metadata objects in the second hierarchical architecture at a corresponding level~~

wherein the first hierarchical architecture and the second hierarchical architecture are implemented according to Object-Oriented class inheritance, wherein each runtime container object at one layer of the first hierarchical architecture functionally interacts with a metadata object at a corresponding layer in the second hierarchical architecture, the dual interaction between the objects in the first hierarchical architecture and the second hierarchical architecture comprising:

invoking application services within a runtime container in the first hierarchical architecture; and

utilizing services performed by a correspondent metadata object in the second hierarchical architecture,

wherein the service provided by a metadata object in the second hierarchical architecture comprises providing state information and context information of the application runtime objects needed by the runtime container object directed to handle at least one of the service requests,

wherein the metadata object comprises object methods to get and set metadata required by the runtime container object,

wherein the runtime container object comprises object methods to invoke the needed service provided by the methods of the metadata object.

24-26. (Canceled).

27. (Original) The method in claim 23, further comprising:

creating new types of runtime containers, or customizing existing containers with incremental features.

28. (Currently Amended) The method in claim 23, further comprising:

creating new levels in the first hierarchical architecture ~~for the runtime container and the~~

second hierarchical architecture the metadata object.

29-42. (Canceled).

43. (Currently Amended) A machine readable medium having instructions stored thereon that when executed by a processor cause a system to:

process service requests and provide application services via a runtime container of a plurality of runtime containers, wherein the plurality of runtime containers are organized in a first hierarchical architecture;

provide metadata on context, state, and/or ~~other~~ runtime information ~~about~~ pertinent to the data and objects being ~~processed~~ handled by the runtime container via a metadata object of a plurality of metadata objects, wherein the plurality of metadata objects are organized in a second hierarchical architecture; and

~~allow each runtime container of the plurality of runtime containers in the first hierarchical architecture to invoke service components within the runtime container and provide state information and context information to the service components based on a meta object of the plurality of metadata objects in the second hierarchical architecture at a corresponding level~~

wherein the first hierarchical architecture and the second hierarchical architecture are implemented according to Object-Oriented class inheritance, wherein each runtime container object at one layer of the first hierarchical architecture functionally interacts with a metadata object at a corresponding layer in the second hierarchical architecture, the dual interaction between the objects in the first hierarchical architecture and the second hierarchical architecture comprising:

invoking application services within a runtime container in the first hierarchical architecture; and

utilizing services performed by a correspondent metadata object in the second hierarchical architecture,

wherein the service provided by a metadata object in the second hierarchical architecture comprises providing state information and context information of the application runtime objects needed by the runtime container object directed to handle at least one of the

service requests,
wherein the metadata object comprises object methods to get and set metadata required
by the runtime container object,
wherein the runtime container object comprises object methods to invoke the needed
service provided by the methods of the metadata object.

44-46. (Canceled).

47. (Original) The machine readable medium of claim 43, further comprising instructions that when executed cause the system to:
create new types of runtime containers, or customize existing containers with incremental features.

48. (Currently Amended) The machine readable medium of claim 43, further comprising instructions that when executed cause the system to:
create new levels in the first hierarchical architecture ~~for the runtime container~~ and the second hierarchical architecture ~~the metadata object.~~

49-64. (Canceled).

65. (New) The system in claim 1, wherein:
at least one of the plurality of runtime containers is a web services container.

66. (New) The system in claim 65, wherein:
at least one of the plurality of runtime containers is a workflow container that inherits from a web services container.

67. (New) The system in claim 1, wherein:
at least one of the plurality of metadata objects is a web services metadata object.

68. (New) The system in claim 67, wherein:

at least one of the plurality of metadata objects is a workflow metadata object that inherits from a web services metadata object.

69. (New) The method in claim 23, wherein:

at least one of the plurality of runtime containers is a web services container.

70. (New) The method in claim 69, wherein:

at least one of the plurality of runtime containers is a workflow container that inherits from a web services container.

71. (New) The method in claim 23, wherein:

at least one of the plurality of metadata objects is a web services metadata object.

72. (New) The method in claim 71, wherein:

at least one of the plurality of metadata objects is a workflow metadata object that inherits from a web services metadata object.

73. (New) The machine readable medium of claim 43, wherein:

at least one of the plurality of runtime containers is a web services container.

74. (New) The machine readable medium of claim 73, wherein:

at least one of the plurality of runtime containers is a workflow container that inherits from a web services container.

75. (New) The machine readable medium of claim 43, wherein:

at least one of the plurality of metadata objects is a web services metadata object.

76. (New) The machine readable medium of claim 75, wherein:

at least one of the plurality of metadata objects is a workflow metadata object that inherits from a web services metadata object.

REMARKS

This proposed claims amendment is for enabling examiner's amendment and also in response to the phone interview between the Examiner Vu, Tuan A. and the Patent Agent Kuiran (Ted) Liu (#60,039) on July 30, 2008 and July 31, 2008.

In view of the above amendments and remarks, it is respectfully submitted that all of the claims now pending in the subject patent application should be allowable, and reconsideration thereof is respectfully requested. The Examiner is respectfully requested to telephone the undersigned if he can assist in any way in expediting issuance of a patent.

Applicant believes that no fee is due with this communication. However, the Commissioner is authorized to charge any underpayment or credit any overpayment to Deposit Account No. 06-1325 for any matter in connection with this reply, including any fee for extension of time, which may be required.

Respectfully submitted,

Date: July 31, 2008

By: /Kuiran (Ted) Liu/
Kuiran (Ted) Liu
Reg. No. 60,039

Customer No. 23910
FLEISLER MEYER LLP
650 California Street, Fourteenth Floor
San Francisco, California 94108
Telephone: (415) 362-3800